An Improved Deep Convolutional Fuzzy System for Classification Problems

Huidong Wang, Jinli Yao
School of Management Science and Engineering
Shandong University of Finance and Economics
Jinan, China
huidong.wang@ia.ac.cn, yjl2mm@126.com

***Abstract*—Fuzzy rule-based classifier is an effective classification algorithm. However, traditional fuzzy system is faced with the challenge of rule explosion and low training speed when dealing with big data problems. In this paper, an improved deep convolutional fuzzy system (DCFS) is proposed for big data classification problems. First, an improved Wang-Mendel (WM) Method is put forward for the training of each sub-fuzzy system. Second, a hierarchical DCFS is constructed for big data classification problems. The system performance is demonstrated via simulation experiments on a couple of classification datasets of varying sizes.**

Keywords—WM Method, deep convolutional fuzzy system, classification, hierarchical fuzzy system

I. INTRODUCTION

Classification is categorizing a sample to one of some predefined classes according to its given feature information. Many supervised machine learning algorithms have been proposed to deal with classification problems [1, 2]. Fuzzy rule-based classifiers [3, 4] are a branch of machine learning technique, which have better interpretability because of its comprehensible IF-THEN rules. Therefore, various fuzzy classifiers are designed in different scenarios [5, 6].

Nowadays, abundant data are available for training fuzzy models. Both the number of features and the number of samples are large in big data [7]. These two significant features have reduced the effectiveness of traditional fuzzy classifiers. On one hand, high-dimensional features will inevitably lead to rule explosion, that is, the number of rules will increase exponentially as the number of features increase. Researchers have developed some feature reduction approaches applied before the training process to solve this problem [8]. On the other hand, previous fuzzy system training algorithms become particularly time-consuming when the number of training samples is large. This is because training algorithms like gradient decent [9,10] and genetic algorithm [11] operate on the whole dataset and the optimal solution is finally found through multiple iterations.

As a hierarchical fuzzy system (FS) [12], deep convolutional fuzzy system (DCFS) [13] shows better performance than simple FS for big data classification problem due to its structure and training algorithm advantages. Its unique structure can overcome the deficiency of simple fuzzy system when facing high-dimensional data. It looks like a pyramid on the whole and is composed of many sub-fuzzy systems. The number of sub-FSs decreases as the layer goes up. The sub-FSs stacked layer-by-layer and the output of a lower layer serves as the input of the higher layer. When a highly dimensional dataset input into DCFS, it is split into several small datasets and delivered to each sub-FS in the first layer. As the data flow up, they are mapped to low-dimensional data gradually through the one-by-one sub-FSs and finally become one output number. This operation manner of DCFS can effectively avoid rule explosion in big data classification problems.

The core training algorithm of DCFS is Wang-Mendel (WM) Method [14, 15], which is a rule extraction approach from data. WM Method is originally designed for simple FSs. Therefore, it can be used to train the rule-base of each sub-FS in DCFS. The prominent advantage of WM Method is that parameters are determined as long as data enter the fuzzy system once. The WM Method no longer needs initiation and parameter fine-tuning. It can reduce the training time to a large extent compared with the iterative algorithms.

However, the working scheme of WM Method can be improved to achieve better classification accuracy. When we apply the WM Method to construct the rule-base for fuzzy systems, the antecedent part is predefined, so it is training for the consequent part. In the WM Method, when one pair of data comes, it is allocated to the rule with the largest firing level and will participate to determine this rule's consequent. Considering the relationship between training process and forward fuzzy inference process [16], this one-data-one-rule strategy is not reasonable. In this paper, we improved the WM Method for training DCFS and applied the improved DCFS to big data classification problems. We also investigated how the number of rules in rule-base can influence classification accuracy.

The rest of this paper is organized as follows. In section 2, we elaborate the improved WM Method and the DCFS classification algorithm. In section 3, the algorithm is tested on several classification datasets. Finally, section 4 draws some conclusions.

II. IMPROVED DCFS FOR CLASSIFICATION PROBLEMS

In this section, we will give details on how to apply a DCFS to deal with big data classification problems. First, steps of the improved WM Method are presented, which is the core training algorithm of DCFS. Next, we introduce the

This work was supported by NSFC Foundation (No.61402260).

978-1-7281-6932-3/20/$31.00 ©2020 IEEE

construction and training process of the whole DCFS. Finally, we discussed the difference between the training of binary and multi-class classification problems.

A. Improved WM Method for Training of Sub-FSs

This part describes the training steps for each sub-FS in DCFS. In essence, it is the training of rule-base which consists of some IF-THEN rules. Each rule in the rule-base takes the following form.

$$R_l\text{: } If\ x_1\ is\ A_{1l}\ and\ ...\ \ x_n\ is\ A_{nl},$$
$$THEN\ y_l(\boldsymbol{x}) = b_{0l} + \sum_{j=1}^{n} b_{jl}x_j\ \ (l = 1, 2, ..., R) \quad (1)$$

$x_j (j = 1, 2, ...n)$ are input variables, which are called antecedents in FSs. $A_{jl} (j = 1, 2, ..n)$ are fuzzy sets corresponding to each input variable. y_l is the output of rule R_l, which is called consequent in FS. In TSK FS [17-19], the consequent is a linear combination of inputs x_1 to x_n. This distinguishes it from Zadeh fuzzy system whose consequents are fuzzy sets [20]. In this paper, the consequent is simplified to a constant b_{0l}.

We have m input-output data pairs $\{x_{i1}, x_{i2}, ..., x_{in};\ y_i\}$ $(i = 1, 2, ..., m)$ in the training set. The number of variables in antecedent part equals to the number of input variables n. Steps of the improved WM Method for training IF-THEN rule base are described as follows.

Stage 1: Input space partition

1) Determine the number of fuzzy sets associated with input variable x_j, denoted as mm_j. The simplest way is that all the variables have the same number of fuzzy sets.

2) Obtain the minimum and maximum value of each input variable x_j from training data, denoted as $\min x_j$ and $\max x_j (j = 1, 2, ..., n)$, respectively.

3) Uniformly distribute mm_j fuzzy sets between $\min x_j$ and $\max x_j$ as Fig.1 shows.

4) Construct IF-THEN rule base. The rule base has $\prod_{j=1}^{n} mm_j$ rules in the form of Eq.(1), where each rule is a combination of the fuzzy sets associated with each variable.

The antecedent part of a rule-base is determined after the above four steps.

Stage 2: Extract rule consequents from data

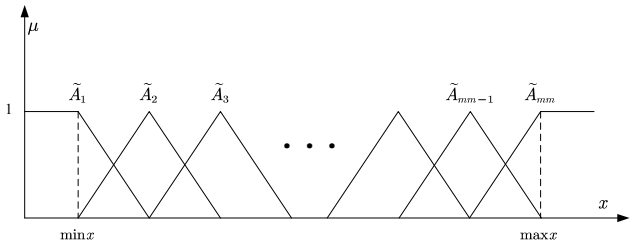

Fig. 1. Distribution of mm fuzzy sets

5) Calculate the firing level of each rule. When an input-output data pair $\{x_{i1}, x_{i2}, ..., x_{in};\ y_i\}$ comes, compute its firing level f_{il} to the rule R_l. One rule's firing level is equal to the product of all the membership grades on the antecedent fuzzy sets of this rule.

6) Find and record the rules being activated and the firing level. We say rule R_l is activated by data pair $\{x_{i1}, x_{i2}, ..., x_{in};\ y_i\}$ if its firing level f_{il} is not equal to zero. There will be $q_1 \cdot q_2 \cdot q_3 ... q_n$ rules being activated when a pair of data comes if the antecedent fuzzy sets distribute like Fig. 1, where $q_j (q_j = 1, 2)$ is the number of fuzzy sets being activated under each input variable. Record the activated rules R_l^*'s firing level f_{il}^* and $f_{il}^* \cdot y_i$.

For the m pairs of data, repeat the above two steps.

7) After going through all the training examples, we can calculate the consequents of the "fired rules". If a rule is activated by at least one pair of data, it is called "fired rule". Suppose Rule R_l^* is a "fired rule", its consequent b_{0l}^* is calculated as follows.

$$b_{0l}^* = \frac{\sum_{k=1}^{p} f_{kl}^* \cdot y_k}{\sum_{k=1}^{q} f_{kl}^*} \quad (2)$$

where k is the subscript of training samples that have fired Rule R_l^*. There are in sum $p (p \leqslant m)$ training samples that have fired Rule R_l^*.

Stage 3: Determine consequents for "not fired rule"

For rules that had not been activated by any data pair in *Stage 2*, their consequents are decided through extrapolation from their neighbors. The neighbors of a rule are the rules whose antecedent fuzzy sets are entirely same as this rule except for one variable. And the only different variable locates in two neighboring fuzzy sets. An example of two neighboring rules is as follows. Here $\tilde{A}_{11}$ is the first fuzzy set of variable x_1 and $\tilde{A}_{12}$ is the second fuzzy set of variable x_1.

$$R_l\text{: } If\ x_1\ is\ A_{11}\ and\ x_2\ is\ A_{21}\ and\ x_3\ is\ A_{31},\ THEN\ y_l = b_{0l} \quad (3)$$

$$R_{l+1}\text{: } If\ x_1\ is\ A_{12}\ and\ x_2\ is\ A_{21}\ and\ x_3\ is\ A_{31},\ THEN\ y_l = b_{0(l+1)} \quad (4)$$

8) Select those rules that have most neighboring "fired rules" from the "not fired rules". These rules are a batch of rules closest to the "fired rules" whose consequents have already been decided.

9) The consequents of these selected "not fired rules" are computed as the arithmetic average of their neighboring rules'.

After *Step 9)*, this selected batch of rules' consequents are decided and then they come into the group of "fired rules". Repeat *Steps 8)* and *9)* until all the rule consequents are obtained.

Compared with the original WM Method, the improvement lies on *Step 5)* and *6)*. In the original WM

Method, when entering a pair of data, only one rule with the biggest firing level is found and recorded. This means one pair of data is only used to calculate one rule consequent. While in our improved method, one pair of data goes to decide all the rule consequents it fires, as is shown in Fig. 2. Without loss of generality, let R_i and R_j represent two different rules, respectively. It can be seen obviously that the data points (represented by multiple crosses) fire both rules and they will participate to decide both these two rules' consequents. Referring to the fuzzy inference process in TSK fuzzy system, our method is consistent with the forward inference that uses weighted average instead of maximum operator. Parameter training is an opposite process to fuzzy inference, hence weighted average is more reasonable to use here.

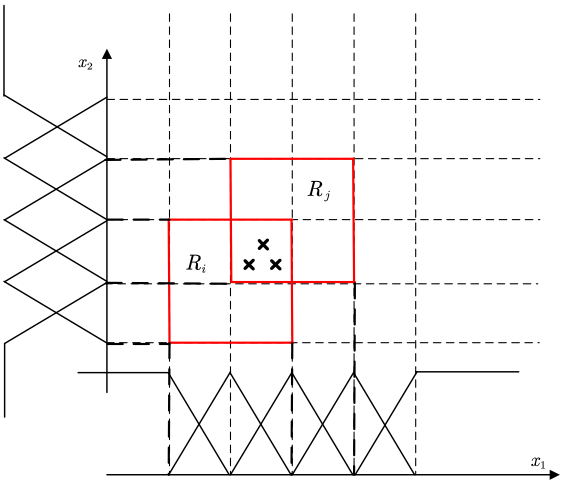

Fig. 2. Working scheme of the improved WM Method.

B. *Construction of the Complete DCFS*

1) Moving window [13]

Moving window is a tool to construct the overall structure of DCFS, which is shown in Fig. 3. The bottom rectangles show how the moving window works. The DCFS is a layer-by-layer hierarchical system and the number of sub-FSs decreases as the layer goes up. Moving window is used to determine the number of sub-FSs in each layer and the number of input variables in every sub-FS.

Moving window has two elements: moving window size and moving scheme. Moving scheme refers to the number of variables moving window moves every time. Fig. 3 is a one-variable-a-time moving window and the moving window size is three. The number of input variables in every sub-fuzzy system equals to moving window size. Once the moving window size and the moving scheme are determined, the number of layers and the number of FSs in each layer can be obtained.

2) Training process of the whole DCFS

In this part, we will discuss how to train the whole DCFS in a layer-by-layer fashion with the improved WM Method. In the training process, the improved WM Method is accompanied by forward fuzzy inference [16, 21, 22].

Assume that the moving window size is three. The whole dataset $\{x_{i1}, x_{i2}, \ldots, x_{in};\ y_i\}$ is divided into n_1 (the number of

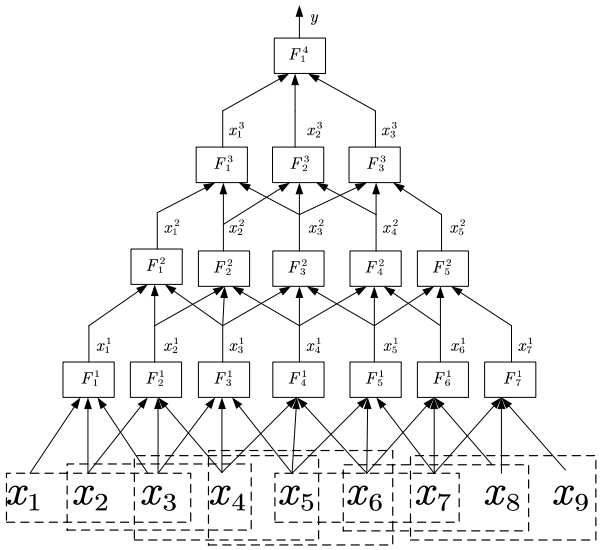

Fig. 3. Overall structure of DCFS.

FSs in the first layer) sub-datasets $\{x_{i1}, x_{i2}, x_{i3};\ y_i\}\{x_{i2}, x_{i3}, x_{i4};\ y_i\}\ldots\{x_{i(n-2)}, x_{i(n-1)}, x_{in};\ y_i\}$ by moving window and sent to the first layer of DCFS, as shown in Fig. 3. Each sub-FS is trained separately by the improved WM Method and the consequents of each sub-FS are obtained. The first layer's training is completed and parameters are fixed. Then we input the feature part of raw data $\{x_{i1}, x_{i2}, x_{i3}\}\{x_{i2}, x_{i3}, x_{i4}\}\ldots\{x_{i(n-2)}, x_{i(n-1)}, x_{in}\}$ into each sub-FS in the first layer, by which we can get first layer's output $\{yy_{11}, yy_{12}, \ldots, yy_{1n_1}\}$ through fuzzy inference. The fuzzy inference process of each sub-FS is similar, here we take the first sub-FS as an example to explain the process. When $\{x_{i1}, x_{i2}, x_{i3}\}$ is input to the first sub-FS in the first layer, we should first calculate the membership grade of the input $\{x_{i1}, x_{i2}, x_{i3}\}$ on the antecedent fuzzy sets of each rule. Then multiply the membership grades and get the firing level of each rule. The final output is aggregation of each rule's consequents with the firing levels as weights. Now we get the output of the first layer.

The first layer's output serves as the input of the second layer. The second layer's parameters are trained by the improved WM Method in the same way. Then the output of the second layer enters the third layer. The parameters are trained layer-by-layer until the top layer.

C. *Binary Classifications and Multi-class Classifications*

Classification problems are divided into binary and multi-class classification according to the number of classes. For binary classification problem, we set the output of positive class as one and the other class as zero. If the DCFS model gives an output that is bigger than 0.5, the sample is classified to the positive class. If the output is less than 0.5, the sample will be classified to the negative class. For multiple classification problems, the one-vs-all strategy is adopted, which means we train a classification model for each class. The certain class is set to one and all the other classes are set to zero when we train classification model for this class.

Finally, samples will be classified to the class whose model gives the biggest output.

III. Simulation Experiments

In this section, the performance of the DCFS will be illustrated via simulation experiments on some big data sets. First, we will introduce the datasets and some preprocessing operations. Then the improved WM Method is tested on a simple dataset. Finally, comprehensive comparison results of DCFSs with the improved WM Method and the original WM Method are given.

A. Datasets

The six classification datasets we used are summarized in Table I. They are all from the UCI Machine Learning Repository. The Iris dataset, which has only 150 examples and four features, is used to compare the improved WM Method with original WM Method in a simple fuzzy system. The DCFS is applied to the other five datasets, using the two training algorithms, respectively.

Seventy percent of each dataset was randomly selected as training set and the remaining thirty percent as test set except dataset "Satellite". The training set and test set of "Satellite" are provided separately, so we did not mix and re-divide it. Experiments were repeated ten times for the other five datasets and the average value was taken as the final result to eliminate the effect of dataset's division on final classification results. All features are numerical and were z-normalized before training.

TABLE I. Classification Datasets

Dataset	No. of samples	No. of features	No. of classes
Iris[a]	150	4	3
Waveform[b]	5000	21	3
Steel[c]	1941	27	7
Yeast[d]	1484	8	10
Clave[e]	10800	16	4
Satellite[f]	6435	36	6

[a.] https://archive.ics.uci.edu/ml/datasets/Iris

[b.] https://archive.ics.uci.edu/ml/datasets/Waveform+Database+Generator+(Version+1)

[c.] https://archive.ics.uci.edu/ml/datasets/Steel+Plates+Faults

[d.] https://archive.ics.uci.edu/ml/datasets/Yeast

[e.] https://archive.ics.uci.edu/ml/datasets/Firm-Teacher_Clave-Direction_Classification

[f.] https://archive.ics.uci.edu/ml/datasets/Statlog+(Landsat+Satellite)

B. Experimental Results

1) Performance of the improved WM Method in simple fuzzy system

Dataset "Iris" has only 150 instances and four features. Hence simple TSK fuzzy system is enough to serve as a classifier. The number of antecedent variables equals to the number of features. The numbers of fuzzy sets mm of each variable are equal and changed from two to twenty. The first subplot in Fig. 4 titled "Iris" shows the classification result of the improved WM Method and original WM Method. We can see from the figure that after mm is larger than ten, the classification accuracy of the improved WM Method keeps better than that of the original WM Method. The best classification accuracy rate of the original WM Method is 0.8660 while the best result of the improved WM Method is 0.8740. Both of them are achieved when mm is equal to 20.

2) Performance of the improved WM Method in DCFS on big data problems

Five datasets are used to illustrate the performance of DCFSs with the improved WM Method and the original WM Method. For the sake of comparison, the same structure parameters are taken for the two DCFSs. The parameters of each datasets are given in Table II. For example, the moving window size for dataset "Waveform" is 3. The moving window moves through 3 variables one time in the first layer and 1 variable in the other layers. There are 4 layers in total, each of them has 7, 5, 3, 1 FSs, respectively. Number of layers and number of fuzzy sets in each layer are determined by moving window size, moving scheme and number of features of each dataset. The structure is designed manually considering convenience of implementation. The classification results are shown in Fig. 4, and Table III shows the best classification accuracy rate of the two algorithms when mm changes.

TABLE II. Parameters of the DCFS

Dataset	Moving window size	Number of variables moved through each time	No. of layers	No. of FSs in each layer
Waveform	3	1st layer: 3 Others: 1	4	7-5-3-1
Steel	3	1st layer: 3 Others: 1	5	9-7-5-3-1
Yeast	3	1	4	6-4-2-1
Clave	1st layer: 4 Others: 3	1st layer: 2 Others: 1	4	7-5-3-1
Satellite	1st layer: 4 Others: 3	1st layer: 4 Others: 1	5	9-7-5-3-1

TABLE III. The Best Classification Accuracy Rate of the Two Algorithms

Datasets	Waveform	Faults	Yeast	Clave	Satellite
Original WM Method	0.8461	0.7203	0.5733	0.7604	0.8775
Improved WM Method	0.8511	0.7266	0.5751	0.7622	0.8790

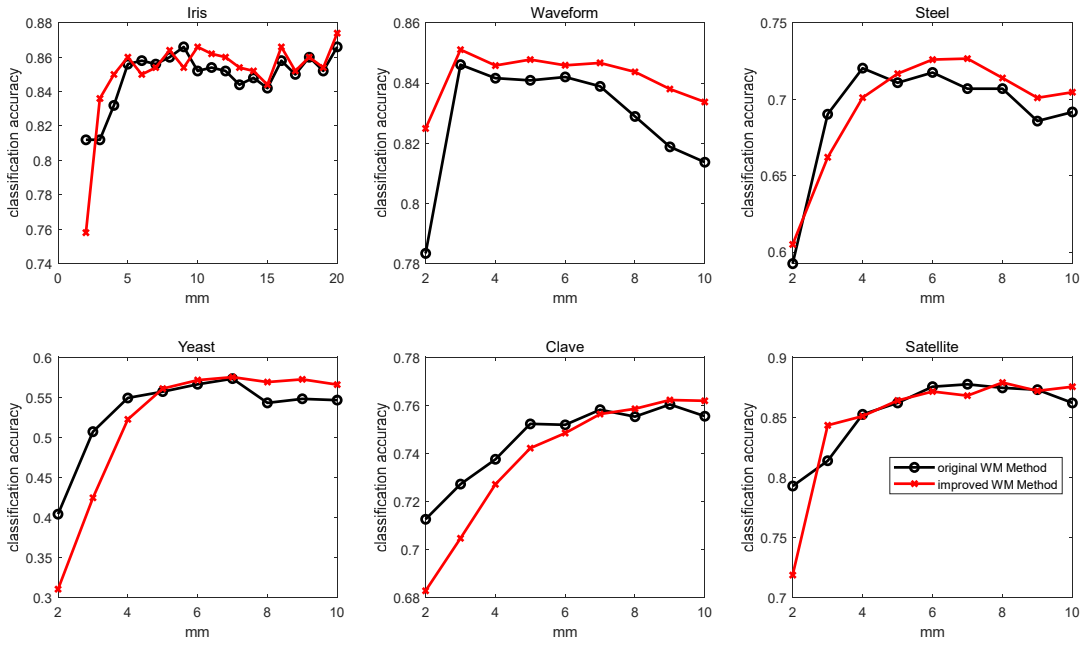

Fig. 4. Classification results with the improved WM Method and the original WM Method.

Exploring the results of Fig. 4 and Table III, We can see that:

a) On dataset "Waveform", DCFS with the improved WM Method always performs better than DCFS with original WM Method, no matter what mm is.

b) As for the other four datasets, the improved WM Method exceeds the original WM Method eventually as mm increases.

c) From Table III we can see that the best classification accuracy rate of improved WM Method is always better than that of original WM Method.

d) In general, as mm increases, both improved and original WM Method performs better. This is in accordance with our expectation because the input space is partitioned more finely as mm increases [23]. There is one exception that classification accuracy increases at first and then decreases when mm continues increasing on the dataset "Waveform".

IV. Conclusions

The traditional fuzzy system has the shortcomings of rule explosion and time-consuming when dealing with big data classification problems. In order to solve these drawbacks, an improved deep convolutional fuzzy system (DCFS) is introduced based on the WM Method. In the improved WM Method, a pair of data can fire multiple fuzzy rules and then participates in determining the rules' consequents. Construction steps of the overall DCFS are given in detail. Simulation experiments are carried out on several representative datasets. Comprehensive results of comparative analysis show the effectiveness and advantages of our method.

In the near future, we will continue to study the impact of system structure on classification accuracy. Without doubt, how to obtain the DCFS's structure automatically is another valuable research problem.

References

[1] J. Maillo, S. García, J. Luengo, F. Herrera, and I. Triguero, "Fast and scalable approaches to accelerate the fuzzy k nearest neighbors classifier for big data," IEEE Trans. Fuzzy Syst., in press.

[2] J. Amezcua and P. Melin, "A new fuzzy learning vector quantization method for classification problems based on a granular approach," Granular Computing, vol. 4, pp. 197–209, April 2019.

[3] Y. Cui, D. Wu, and J. Huang, "Optimize TSK fuzzy systems for classification problems: Mini-batch gradient descent with uniform regularization and batch normalization," IEEE Trans. Fuzzy Syst., in press.

[4] Y. Deng, Z. Ren, Y. Kong, F. Bao, and Q. Dai, "A hierarchical fused fuzzy deep neural network for data classification," IEEE Trans. Fuzzy Syst., vol. 25, pp. 1006–1012, June 2016.

[5] B. Soua, A. Borgi, and M. Tagina, "An ensemble method for fuzzy rule-based classification systems," Knowl. Inf. Syst., vol. 36, pp. 385–410, August 2013.

[6] H.-X. Li, Y. Wang, and G. Zhang, "Probabilistic fuzzy classification for stochastic data," IEEE Trans. Fuzzy Syst., vol. 25, pp. 1391–1402, December 2017.

[7] A. McAfee, E. Brynjolfsson, T. H. Davenport, D. Patil, and D. Barton,"Big data: the management revolution," Harv. Bus. Rev., vol. 90, pp. 60–68, October 2012.

[8] H. Zhao, P. Wang, Q. Hu, and P. Zhu, "Fuzzy rough set based feature selection for large-scale hierarchical classification," IEEE Trans. Fuzzy Syst., vol. 27, pp. 1891–1903, October 2019.

[9] L.-X. Wang and J. M. Mendel, "Back-propagation fuzzy system as nonlinear dynamic system identifiers," in Proc. IEEE Int'l Conf. on Fuzzy Systems, San Diego, CA, 1992, pp. 1409–1418.

[10] J.-S. R. Jang, “ANFIS: adaptive-network-based fuzzy inference system,” IEEE Trans. on Systems, Man, and Cybern., vol. 23, pp. 665–685, May 1993.

[11] D. Wu and W. Wan Tan, “Genetic learning and performance evaluation of interval type-2 fuzzy logic controllers,” Eng. Appl. Artif. Intell., vol. 19, pp. 829–841, December 2006.

[12] G.V.S. Raju, J. Zhou and R. A. Kisner, "Hierarchical fuzzy control", Int. J. Contr., vol. 54, pp. 1201-1216, March 1991.

[13] L.-X. Wang, “Fast training algorithms for deep convolutional fuzzy systems with application to stock index prediction,” IEEE Trans. Fuzzy Syst., in press.

[14] L.-X. Wang and J.M. Mendel, “Generating fuzzy rules by learning from examples,” IEEE Trans. on Systems, Man, and Cybern., vol. 22, pp. 1414-1427, November 1992.

[15] L.-X. Wang, “The WM method completed: A flexible fuzzy system approach to data mining,” IEEE Trans. Fuzzy Syst., vol. 11, pp. 768-782, December 2003.

[16] J. M. Mendel, Uncertain rule-based fuzzy systems: introduction and new directions, 2nd ed. Springer, 2017.

[17] A. Nguyen, T. Taniguchi, L. Eciolaza, V. Campos, R. Palhares, and M. Sugeno, “Fuzzy control systems: Past, present and future,” IEEE Comput. Intell. Mag., vol. 14, pp. 56–68, January 2019.

[18] D. Wu and J. M. Mendel, “Recommendations on designing practical interval type-2 fuzzy systems,” Eng. Appl. Artif. Intell., vol. 85, pp. 182–193, October 2019.

[19] D. Wu, Y. Yuan, J. Huang, and Y. Tan, “Optimize TSK fuzzy systems for big data regression problems: Mini-batch gradient descent with regularization, DropRule and AdaBound (MBGD-RDA),” IEEE Trans. Fuzzy Syst., in press.

[20] D. Wu, C.-T. Lin, J. Huang, and Z. Zeng, “On the functional equivalence of TSK fuzzy systems to neural networks, mixture of experts, CART, and stacking ensemble regression,” IEEE Trans. Fuzzy Syst., in press.

[21] L.-X. Wang, A Course in Fuzzy Systems and Control, Englewood Cliffs, NJ, USA:Prentice-Hall, 1997.

[22] J. M. Mendel and D. Wu, Perceptual Computing: Aiding People in Making Subjective Judgments. Wiley-IEEE Press, Hoboken, NJ, 2010.

[23] J. M. Mendel, “Explaining the performance potential of rule-based fuzzy systems as a greater sculpting of the state space,” IEEE Trans. Fuzzy Syst., vol. 26, pp. 2362–2373, August 2018.